STL Vector Sequential Container

- A `std::vector` is a dynamic array that can grow & shrink at the end efficiently
  - e.g., it provides (pre/re) allocation, indexed storage, `push_back()` & `pop_back()`
- Supports *random access* iterators
- Similar to—but more powerful than—built-in C/C++ arrays

```
template <typename T,
          typename Allocator =
             allocator<T>>
class vector;
```

1. **T**: Datatype of the elements that can be stored in the vector.

2. **Allocator**: This is the allocator object used to define the storage allocation model
   - By default, the Allocator class template from <memory> for type T is used, which defines the simplest memory allocation model and is value-independent.

See www.cplusplus.com/reference/vector/vector

STL Vector Sequential Container Examples

```
int main() {
    vector<int> v(3);
    v[0] = 7; v[1] = v[0] + 3; v[2] = v[0] + v[1];

    for (int i=0; i< v.size() ; ++i) cout << v[i] << ' ';
    cout << endl;

    v.resize(0);

    for (int i=0; i < 7; ++i) v.push_back(i);

    for (int i : v) cout << i << ' ';
    cout << endl;
    return 0;
}
```

See github.com/douglascraigschmidt/CPlusPlus/tree/master/STL/S-03/3.3